**CENTRALIS HATARELOSZLAS TETEL**

#Kockadobás szimuláción keresztul CHT szemléltetés

n<-1000 #kisérlet száma

k<-10 # 1 kísérleten belül a dobások

# EGYENLETES ELOSZLAS # EXPONENCIALIS ELOSZLAS

x<-round(runif(n\*k,1,6)) lambda<-2

x[1:100] x<-rexp(n\*k,rate=lambda);x

# GAMMA ELOSZLAS # NORMALIS ELOSZLAS

b<-1 #skála m<-2 #várható érték

a<-1 #alak sd<-10 #szórás

x<-rgamma(n,a,b); x<-rnorm(n,m,sd);

# WEIBULL ELOSZLAS

b<-1 #skála

a<-1 #alak

x<-rweibull(n,a,b);

# CAUCHY ELOSZLAS

l<-0 #location

s<-1 #scale

x<-rcauchy(n,l,s);x

y<-x[x>-10 & x<10]

X<-matrix(y,ncol=k)

X<-matrix(x,ncol=k) #n: sor (kísérletek), k:oszlop (dobások)

head(X)

col\_sum<-apply(X,1,sum);col\_sum #minden kísérlet dobásösszege (1: sor, 2: oszlop)

par(mfrow=c(1,2))

hist(col\_sum,freq = F) #normálisra hasonló

#Alkalmazzuk a CHT-t

mean(col\_sum) #34.856 >> elméleti: 35=10\*3.5

sd(col\_sum) #4.764448 >> elméleti: 4.564355=sqrt((6-11))^2/12)\*sqrt(10)

Z<-(col\_sum-mean(col\_sum))/sd(col\_sum);Z #Standardizálás

mean(Z) #3.479812e-16 >> elméleti: 0

sd(Z) #1 >> elméleti: 1

hist(Z,freq=F) #Standard normal

lines(seq(-3,3,0.05),dnorm(seq(-3,3,0.05)),col="red") #Standard normalis >>> CHT

**PARAMETERBECSLES MAXIMUM LIKELIHOOD MODSZERREL**

#Exponenciális eloszlás lambda paraméterének becslése

n<-1000

l<-3 #lambda, ezt fogjuk becsülni!!!

x<-rexp(n,rate=l)

#Log-likelihood függvény:

exp\_loglik<-function(y,par){

loglik<-sum(dexp(y,rate=par,log=TRUE))

return(-loglik)

}

#Mgj: azért vesszük a (-1)-szeresét, mert az optimalizáló algoritmus

#minimum keresésre van megirva

#Optimalizálás: nlm(f= optimalizálandó függvény, p= paraméter kezdeti értéke)

#Mgj: Érzékeny a kiinduló érték választásra, hogy választunk??

opt<-nlm(f=exp\_loglik, p=1, y=x);opt #kiinduló érték lambda=1

opt$estimate #paraméter becslése

#Mit csináltunk?

par(mfrow=c(1,3))

hist(x,freq = F,main="Exp(l) sürüségfv")

points(seq(0,2,0.05),dexp(seq(0,2,0.05),rate=l),type="l")

hist(x,freq = F,main="Exp(ML becsléssel kapott lambda) sürüségfv")

points(seq(0,2,0.05),dexp(seq(0,2,0.05),rate=opt$estimate),type="l", col="red")

hist(x,freq = F,main="Kettő együtt")

points(seq(0,2,0.05),dexp(seq(0,2,0.05),rate=l),type="l")

points(seq(0,2,0.05),dexp(seq(0,2,0.05),rate=opt$estimate),type="p", col="red")

##########################################################################

#Normális eloszlás m- várható érték és s - szórás paraméterének becslése

n<-1000

m<-3 #várható érték, ezt fogjuk becsülni!!!

s<-2 #vszórás, ezt fogjuk becsülni!!!

x<-rnorm(n,mean=m,sd=s)

#Log-likelihood függvény:

norm\_loglik<-function(y,par){

loglik<-sum(dnorm(y,mean=par[1], sd=par[2],log=TRUE))

return(-loglik)

}

opt<-nlm(f=norm\_loglik, p=c(1,1), y=x);opt #kiinduló érték m=1, sd=1

opt$estimate #paraméter becslése

-m-s-3

m+s+3

#Mit csináltunk?

par(mfrow=c(1,2))

hist(x,freq = F,main="Norm(m,s) sürüségfv")

points(seq(-8,8,0.1),dnorm(seq(-8,8,0.1),mean=m,sd=s),type="l")

hist(x,freq = F,main="Norm(ML becsléssel kapott m és s) sürüségfv")

points(seq(-8,8,0.1),dnorm(seq(-8,8,0.1),mean=opt$estimate[1],sd=opt$estimate[2]),type="l", col="red")

##########################################################################

#Gamma eloszlás a - hely, lambda - skála paraméterének becslése

n<-1000

a<-5 #alak (shape), ezt fogjuk becsülni!!!

l<-3 #skála (scale), ezt fogjuk becsülni!!!

x<-rgamma(n,shape=a,scale=l)

#Log-likelihood függvény:

gamma\_loglik<-function(y,par){

loglik<-sum(dgamma(y,shape=par[1], scale=par[2],log=TRUE))

return(-loglik)

}

opt<-nlm(f=gamma\_loglik, p=c(1,1), y=x);opt #kiinduló érték a=1, l=1

opt$estimate #paraméter becslése

#Mit csináltunk?

par(mfrow=c(1,2))

hist(x,freq = F,main="Gamma(a,l) sürüségfv")

points(seq(0,a^2,0.1),dgamma(seq(0,a^2,0.1),shape=a,scale=l),type="l")

hist(x,freq = F,main="Norm(ML becsléssel kapott a és l) sürüségfv")

points(seq(0,a^2,0.1),dgamma(seq(0,a^2,0.1),shape=opt$estimate[1],scale=opt$estimate[2]),type="l", col="red")

##########################################################################

#Weibull eloszlás c=0 -eltolás, a-alak (shape), b -skála (scale), paraméterrel

#1. eset: Exponenciális

a<-0.5 #alak (shape), ezt fogjuk becsülni!!!

b<-1 # skála (scale), ezt fogjuk becsülni!!!

#2. eset: Rayleigh

a<-2 #alak (shape), ezt fogjuk becsülni!!!

b<-1 # skála (scale), ezt fogjuk becsülni!!!

#1. eset: Normális

a<-3.57 #alak (shape), ezt fogjuk becsülni!!!

b<-1 # skála (scale), ezt fogjuk becsülni!!!

x<-rweibull(n,shape=a,scale=b)

#Log-likelihood függvény:

weibull\_loglik<-function(y,par){

loglik<-sum(dweibull(y,shape=par[1], scale=par[2],log=TRUE))

return(-loglik)

}

opt<-nlm(f=weibull\_loglik, p=c(2,2), y=x);opt #kiinduló érték l=1, s=1

opt$estimate #paraméter becslése

#Mit csináltunk?

par(mfrow=c(1,2))

hist(x,freq = F,main="Weibull(a,b) sürüségfv")

points(seq(0,25,0.1),dweibull(seq(0,25,0.1),shape=a,scale=b),type="l")

hist(x,freq = F,main="Weibull(ML becsléssel kapott a és b) sürüségfv")

points(seq(0,25,0.1),dweibull(seq(0,25,0.1),shape=opt$estimate[1],scale=opt$estimate[2]),type="l", col="red")

##########################################################################

#Cauchy eloszlas

n<-10000

l<-0 #location ezt fogjuk becsülni

s<-1 #scale ezt fogjuk becsülni

x<-rcauchy(n,l,s)

z<-x[x>-10 & x<10] #szûrés

#Log-likelihood függvény:

cauchy\_loglik<-function(y,par){

loglik<-sum(dcauchy(y,location=par[1], scale=par[2],log=TRUE))

return(-loglik)

}

opt<-nlm(f=cauchy\_loglik, p=c(0,1), y=z);opt #kiinduló érték l=1, s=1

opt$estimate #paraméter becslése

#Mit csináltunk?

par(mfrow=c(1,2))

hist(z,freq = F,main="Cauchy(l,s) sürüségfv")

points(seq(-8,8,0.1),dcauchy(seq(-8,8,0.1),l,s),type="l")

hist(z,freq = F,main="Cauchy(ML becsléssel kapott l és s) sürüségfv")

points(seq(-8,8,0.1),dcauchy(seq(-8,8,0.1),opt$estimate[1],opt$estimate[2]),type="l", col="red")

##########################################################################

#Gamma eloszlás a - hely, lambda - skála paraméterének becslése

n<-1000

a<-5 #alak (shape), ezt fogjuk becsülni!!!

l<-3 #skála (scale), ezt fogjuk becsülni!!!

x<-rgamma(n,shape=a,scale=l)

#Log-likelihood függvény:

gamma\_loglik<-function(y,par){

loglik<-sum(dgamma(y,shape=par[1], scale=par[2],log=TRUE))

return(-loglik)

}

opt<-nlm(f=gamma\_loglik, p=c(1,1), y=x);opt #kiinduló érték a=1, l=1

opt$estimate #paraméter becslése

#Mit csináltunk?

par(mfrow=c(1,2))

hist(x,freq = F,main="Gamma(a,l) sürüségfv")

points(seq(0,a^2,0.1),dgamma(seq(0,a^2,0.1),shape=a,scale=l),type="l")

hist(x,freq = F,main="Gamma(ML becsléssel kapott a és l) sürüségfv")

points(seq(0,a^2,0.1),dgamma(seq(0,a^2,0.1),shape=opt$estimate[1],scale=opt$estimate[2]),type="l", col="red")

**VELETLENSZAM GENERALAS**

#Véletlenszám generáló függvényünk: kongruenciális

velszam.gen<-function(N,a,c,m){

X = matrix(0, ncol=2, nrow=N)

x = 1

for(i in 1:N) {

x = (a\*x+c) %%m

X[i,1] = x

X[i,2] = x/m} #ha [0,1]-beli elemet szeretnénk

X

}

velszam.gen(15,7,0,13) #órai példa

x<-velszam.gen(15,7,0,13)[,2] #N=100,1000

plot(seq(0,1,length.out=length(x)),x)

velszam.gen(15,65539,0,2^31) #RANDU generátor

x<-velszam.gen(15,65539,0,2^31)[,2] #N=100,1000,10000

plot(seq(0,1,length.out=length(x)),x)

#RANDU kongruenciális generátor

randu #beépített példa 400\*3

r=cbind(randu[,1],randu[,2],randu[,3]);r

hist(randu[,1])

plot(randu[,1]) #ez még egyenletes

plot(randu[,3]-6\*randu[,2]+9\*randu[,1]) #igy már nem egyenletes

??plot3d

install.packages("rgl") #MENÜ

library(rgl) #LOAD

plot3d(randu[,1],randu[,2],randu[,3])

rgl.viewpoint(theta = -3.8, phi = 3.8, fov = 0, zoom = 0.7) #•Ebből a szögből nézve

plot3d(randu[,1],randu[,2],randu[,3])

###########################################################

#Egyenletes eloszlásból generáljunk Exp(lambda)-t

lambda<-2

N<-1000

exp\_gen<-function(N,lambda){

U=runif(N) #N db egyenletes generálása

X=-1/lambda\*log(1-U) #Elkészítjük az N db exponenciálist

Y=rexp(N,lambda) #Exp generálása

par(mfrow=c(1,2))

hist(X,freq=F,main="Exp from Uniform")

points(seq(0,10,0.01),dexp(seq(0,10,0.01),lambda),type="l",col="red")

hist(Y,freq=F,main="Exp from R")

points(seq(0,10,0.01),dexp(seq(0,10,0.01),lambda),type="l",col="red")

}

exp\_gen(10000,1)

###########################################################

#Egyenletes eloszlásból generáljunk Gamma(N,lambda)-t

gamma\_gen<-function(n,N,lambda){

Z = matrix(0, ncol=2, nrow=n)

for (i in 1:n) {

U=runif(N) #N db egyenletest generálunk

Z[i,1]=-sum(log(1-U)/lambda)} #1 db gammáért

Z[,2]<-rgamma(n,shape=N,rate=lambda)

par(mfrow=c(1,2))

hist(Z[,1],freq=F,main="Gamma from Uniform")

points(seq(0,10,0.01),dgamma(seq(0,10,0.01),shape=N,rate=lambda),type="l",col="red")

hist(Z[,2],freq=F,main="Gamma from R")

points(seq(0,10,0.01),dgamma(seq(0,10,0.01),shape=N,rate=lambda),type="l",col="red")

}

gamma\_gen(5000,10,2)

##############################################################

#Egyenletes eloszlasbol generaljunk cauchy(l,s)-t

cauchy\_gen<-function(N,l,s){

U=runif(N) # N db egyenletes generalasa

X=s\*tan(pi\*(U-1/2))+l # N db cauchy-hoz

par(mfrow=c(1,2))

x<-X[X>-10&X<10]

hist(x,freq=F,main="Cauchy from Uniform")

points(seq(-10,10,0.05),dcauchy(seq(-10,10,0.05),location=l,scale=s),type="l",col="red")

Y=rcauchy(N,location=l, scale=s) # cacuhy generalasa

y<-Y[Y>-10&Y<10]

hist(y,freq=F,main="Cauchy from R")

points(seq(-10,10,0.05),dcauchy(seq(-10,10,0.05),location=l,scale=s),type="l",col="red")

}

cauchy\_gen(1000,2,1)

#############################################################

#Egyenletes eloszlasbol generaljunk Weibull(a=shape,b=scale)-t

weibull\_gen<-function(N,a,b){

U=runif(N) # N db egyenletes generalasa

X=b\*(-log(1-U))^(1/a) # N db weibull-hoz

par(mfrow=c(1,2))

hist(X,freq=F,main="Weibull from Uniform")

points(seq(-N,N,0.01),dweibull(seq(-N,N,0.01),shape=a,scale=b),type="l",col="red")

Y=rweibull(N,shape=a,scale=b) # weibull generalasa

hist(Y,freq=F,main="Weibull from R")

points(seq(-N,N,0.01),dweibull(seq(-N,N,0.01),shape=a,scale=b),type="l",col="red")

}

weibull\_gen(1000,2,1)

##############################################################

#Normáis eloszlás generálása egyenletesből

#CHT

u<-runif(12,0,1);u

z<-sum(u)-6;z

norm\_gen<-function(n){

z<-matrix(0,ncol=1,nrow=n)

for (i in 1:n){

u<-runif(12,0,1) #12 db egyenletest generálunk

z[i]<-sum(u)-6 #1 db normálisért

}

par(mfrow=c(1,2))

hist(z,freq=F,main="Normal from CHT")

points(seq(-3,3,0.01),dnorm(seq(-3,3,0.01)),type="l",col="red")

y<-rnorm(n)

hist(y,freq=F,main="Normal from R")

points(seq(-3,3,0.01),dnorm(seq(-3,3,0.01)),type="l",col="red")

}

norm\_gen(1000)

##############################################################

#Box-Müller algoritmus

box\_muller<-function(n){

z = matrix(0, ncol=2, nrow=n)

for (i in 1:n){

u<-runif(2,0,1) #2 db egyenletest generálunk

z[i,1]<-sqrt(-2\*log(u[1]))\*cos(2\*pi\*u[2]) # 1-1 db normálisért

z[i,2]<-sqrt(-2\*log(u[1]))\*sin(2\*pi\*u[2])

}

x<-cbind(z[,1],z[,2]) #A mátrix 2 oszlopát egyesítjük.

par(mfrow=c(1,2))

hist(x,freq=F,main="With Box-Müller")

points(seq(-3,3,0.01),dnorm(seq(-3,3,0.01)),type="l",col="red")

y<-rnorm(2\*n)

hist(y,freq=F,main="Normal from R")

points(seq(-3,3,0.01),dnorm(seq(-3,3,0.01)),type="l",col="red")

}

box\_muller(10000)

##############################################################

#Stabilis generator

install.packages("stabledist") #MENÜ

library(stabledist) #LOAD

stabilis\_gen<-function(n){

alpha=2; #alpha ]0;2] és alpha!=1

lambda=1;

beta = 1; #ferdeség: [-1; 1]

z = array(n)

for (i in 1:n){

xszi<-runif(1,-pi/2,pi/2) #1 db egyenletes

eta<-rexp(1,lambda)

z[i]<-(sin(alpha\*xszi)/(cos(xszi)^(1/alpha)))\*((cos((1-alpha)\*xszi)/eta))^((1-alpha)/alpha)

}

par(mfrow=c(1,2))

hist(z,freq=F,main="Zolotarev-féle")

points(seq(-6,6,0.01),dstable(seq(-6,6,0.01),alpha,beta),type="l",col="red")

y<-rstable(n,alpha,beta) #gamma a scale, delta shift lenne még

hist(y,freq=F,main="Stabilis R hívással")

points(seq(-6,6,0.01),dstable(seq(-6,6,0.01),alpha,beta),type="l",col="red")

}

stabilis\_gen(10000)

##############################################################

#Inverz fgv módszer általánosan

#1. állitás Exp(lambda) #ennek normál eloszlásra kell hasonlítani

n<-1000

lambda<-2

rexp(n, rate=lambda)

x<-pexp(rexp(n, rate=lambda), rate=lambda)

par(mfrow=c(1,2))

plot(1:n,x)

hist(x)

#1. állítás: norm(mean,sd) #ennek normál eloszlásra kell hasonlítani

n<-1000

m<-2

s<-1

x<-pnorm(rnorm(n,m,s),m,s)

par(mfrow=c(1,2))

plot(1:n,x)

hist(x,freq=F)

#2. állítás exp(lambda)

m<-1000

lambda<-2

x<-qexp(runif(n),rate=lambda)

par(mfrow=c(1,2))

hist(x,freq=F)

points(seq(0,4,0.05),dexp(seq(0,4,0.05),rate=lambda), type="l", col="red")

x<-qexp(runif(n),rate=lambda)

y<-qexp(runif(n),rate=lambda)

plot(x,y)

#2. állítás: Cauchy(1,s)

n<-1000

l<-0

s<-1

runif(n)

x<-qcauchy(runif(n),location=1,scale=s)

y<-x[x>-10 & x<10]

plot(y)

hist(y,freq=F)

points(seq(-10,10,0.05),dcauchy(seq(-10,10,0.05),location=l,scale=s),type="l", col="red")

#2. állítás: norm(mean,sd)

n<-1000

m<-2

s<-1

rnorm(n,m,s)

par(mfrow=c(1,2))

x<-qnorm(runif(n),m,s)

hist(x,freq=F)

plot(x)

**REGRESSZIO**

#### R-ben Lineáris regresszió #####

x<-runif(10,-5,5) #Magyarázó változó (independent)

y<-x+rnorm(10,2,1)#Függő változó (dependent)

#Fitting a linear model

model<-lm(y~x);model #egyenes

#Summary of the model

summary(model) #Mi micsoda?

#Grafika

plot(x,y) #pontpárok kirajzolása

abline(model,col="red",lwd=2,lty=2) #illesztett egyenes rajzolása

#Diagnostic plots: NOT a plot of the model itself!

plot(model) #Residual plot; QQplot

#Előrejelzés

model$coefficients

model$coefficients[1]+model$coefficients[2]\* 2 #előrejelzés Y=Intercept+Slope\*30

#VAGY

predict(model,newdata=data.frame(x=2)) #előrejelzés X=30-ra

#ábrán

plot(x,y)

abline(model,col="red",lwd=2,lty=2)

points(2,model$coefficients[1]+model$coefficients[2]\*2,col="blue",lwd=3,pch=2)

#Hiba

predict(model) #b[1]+b[2]\*x: becsült értékek

y-predict(model) # tényleges - becslés: a hibáják

sum((y-predict(model))^2) #hibák négyzetösszege

#modell hibája: Residual standard error

summary(model)

anova(model)

sqrt(sum((y-predict(model))^2)/(10-2)) #10-2:length(x)-tagok száma (konst,x)

plot(x,y,xlim=c(-5,5),ylim=c(-5,5))

abline(model,col="red",lwd=2,lty=2)

p<-model$coefficients[1]+model$coefficients[2]\*x

e<-abs(y-p)

rect(x,y,x+e,p)

#Konfidencia intervallum

confint(model,level=0.95)

#Egy új x értékre becslést és konf int adni:

predict(model,newdata=data.frame(x=2),interval = "confidence", level = 0.95)

#For a given value of x, the interval estimate for the mean of the dependent variable, —y , is called the confidence interval.

newx <- seq(-5, 5, by=0.1);newx

confint<-predict(model,newdata=data.frame(x=newx),interval = "confidence");confint

par(mfrow=c(1,1))

plot(x,y, main="Confidence interval")

lines(newx, confint[,1], col="red", lty=1)

lines(newx, confint[,2], col="blue", lty=2)

lines(newx, confint[,3], col="blue", lty=2)

#Egyéb kétváltozós modellek

#1.Kvadratikus

x<-runif(10,-5,5)

y<-x^2+rnorm(10,2,1)

plot(x,y)

#Fitting a kvadratic model

model<-lm(y~x+I(x^2));model #parabola

summary(model)

#Reduced model

model<-lm(y~I(x^2));model #parabola

summary(model)

#Parabola illesztés

f<-model$coefficients[1]+model$coefficients[2]\*seq(-5,5,0.1)^2

points(seq(-5,5,0.1),f,type="l",col="red")

#2. Logaritmikus

x<-runif(10,0,10)

y<-x+log(x)+rnorm(10,3,2)

plot(x,y)

#Fitting a kvadratic model

model<-lm(y~x+I(log(x)));model #parabola

summary(model)

#Parabola illesztés

f<-model$coefficients[1]+model$coefficients[2]\*seq(0,10,0.1)+model$coefficients[3]\*log(seq(0,10,0.1))

points(seq(0,10,0.1),f,type="l",col="red")

#Többváltozós regresszió

longley

attach(longley)

head(longley)

#Full model: Residual standard error: 0.3049; Adjusted R-squared: 0.9925

model<-lm(Employed ~ GNP.deflator + GNP + Unemployed + Armed.Forces + Population + Year)

summary(model)

#Reduced model1:Residual standard error: 0.2897; Adjusted R-squared: 0.9932

model<-lm(Employed ~ GNP + Unemployed + Armed.Forces + Population + Year)

summary(model)

#Residual standard error: 0.2794; Adjusted R-squared: 0.9937

model<-lm(Employed ~ GNP + Unemployed + Armed.Forces + Year)

summary(model)

#Reduced model3: Residual standard error: 0.3321; Adjusted R-squared: 0.9911

model<-lm(Employed ~ Unemployed + Armed.Forces + Year)

summary(model)

#Reduced model3: Residual standard error: 0.5017; Adjusted R-squared: 0.9796

model<-lm(Employed ~ Unemployed + Year)

summary(model)

#Residual standard error (x): HIBA >> MIN

#Adjusted R-squared (y): magyarázó erő >> MAX

x<-c(0.3049,0.2897,0.2794,0.3321,0.5017)

y<-c(0.9925,0.9932,0.9937,0.9911,0.9796)

plot(x,y,type="b",xlab="Residual standard error",ylab=" Adjusted R-squared")

text(x[1],y[1],"FM",c(1,1))

text(x[2],y[2],"RM1",c(1,1))

text(x[3],y[3],"RM2",c(0,0))

text(x[4],y[4],"RM3",c(1,1))

text(x[5],y[5],"RM4",c(1,1))

#BEST!!! >> Reduced model2:

points(min(x),max(y),col="red",pch="\*",cex=2)

##############################################################

#PELDAFELADAT

iskolazottsag<-c(12.6,12.4,11.6,10.4,4.4) #Magyarázó változó (independent)

varhato\_elettartam<-c(81.1,78.5,75.4,74,65.4) #Függő változó (dependent)

#Fitting a linear model

model<-lm(varhato\_elettartam~iskolazottsag);model #egyenes

#Intercept: x=0-hoz tartozó y érték, azaz ha valaki

#0 évet tanul, akkor ennyi a várható élettartama (beta0)

#Másik érték: ha 1 évvel többet tanul, akkor annyi a növekedett

#élettartam. Azaz ha az x értékét egyel növeljük akkor az

#y annyival növekszik

#Summary of the model

summary(model)

elorejelzes<-predict(model,newdata=data.frame(iskolazottsag=12)) #előrejelzés

#Grafika

plot(iskolazottsag,varhato\_elettartam) #pontpárok kirajzolása

abline(model,col="red") #ilesztett egyenes rajzolása

points(12,elorejelzes,col="blue",lwd=3,pch=2) #előrejelzett érték jelölése

**PORTFOLIO**

#Adatok generálása

X<-matrix(0,ncol=12,nrow=11)

#Kis hozam, Kis kockázat - Nagy hozam, Nagy kockázat

for (i in 1:10){

set.seed(i)

X[i,]<-rnorm(12,mean=i,sd=i)}

#Kockázatmentes banki kötvény

X[11,]<-rep(0.05,n=12)

X<-t(X)

head(X)

colnames(X)<-paste(1:11,'termék',sep='.')

row.names(X)<-paste(1:12,"hó",sep=".")

head(X)

#Éves hozam

r<-colMeans(X);r

par(mfrow=c(1,2))

plot(1:11,r,xlab="Befektetési eszközök", ylab="Éves hozam")

#Szórás

sigma<-NULL

for (i in 1:11){sigma[i]<-sd(X[,i])}

names(sigma)<-names(r)

plot(1:11,sigma,xlab="Befektetési eszközök", ylab="Szórás")

#Fektessünk be!

par(mfrow=c(1,1))

plot(0:10,0:10,type="n",xlab="Kockázat",ylab="Hozam",main="Lehetséges portfóliók")

#1. Csak a legnagyobb hozamúba fektessük a pénzünket.

which.max(r)

which.max(kock) #egyben a legnagyobb kockázatú is

w1<-c(0,0,0,0,0,0,1,0,0,0,0)

h1<-sum(r\*w1);h1

k1<-sqrt(w1%\*%cov(X)%\*%w1);k1

points(k1,h1,col="blue",pch=16)

text(k1,h1,"1",c(1,0))

#2. Egyformán fektessük be pénzünket

w2<-rep(1/11,11)

h2<-sum(r\*w2);h2

k2<-sqrt(w2%\*%cov(X)%\*%w2);k2

points(k2,h2,col="blue",pch=16)

text(k2,h2,"2",c(1,0))

#3. Csak kötvénybe fektessük be pénzünket

w3<-c(rep(0,10),1)

h3<-sum(r\*w3);h3

k3<-sqrt(w3%\*%cov(X)%\*%w3);k3

points(k3,h3,col="blue",pch=16)

text(k3,h3,"3",c(1,0))

#4. Kaptunk egy tippet

w4<-c(0, 0, 0, 0.1, 0, 0.2, 0.25, 0, 0.25, 0.2, 0 )

sum(w4)

h4<-sum(r\*w4);h4

k4<-sqrt(w4%\*%cov(X)%\*%w4);k4

points(k4,h4,col="blue",pch=16)

text(k4,h4,"4",c(1,0))

#5. Egyéni ötlet:

w5<-c()

sum(w5)

h5<-sum(r\*w5);h5

k5<-sqrt(w5%\*%cov(X)%\*%w5);k5

points(k5,h5,col="green",pch=16)

install.packages('quadprog')

library(quadprog)

#Mi lesz az optimális portólió?

#Azaz a legnagyobb hozamú és legkisebb kockázatú?

portfolio\_fv <- function(data,r)

{

n <- ncol(data) #adatsor oszlopainak a száma

Dmat <- cov(data) ### kovarianciamátrix

dvec <- rep(0, times=n) ### nullvektor n elemű

Amat <- cbind(colMeans(data), rep(1, n),diag(n)) #oszloponként összerakja a mátrixot

bvec <- c(r, 1, rep(0, times=n)) #vektort állít elő

meq <- 2

portfolio = solve.QP(Dmat, dvec, Amat, bvec, meq) #optimalizáció!!!

weights = round(portfolio$solution, digits = 4) #megoldás kerekítése-->> SÚLYOK

names(weights) = colnames(data)

list(weights = weights,risk = portfolio$value,return = r)

}

data<-X[,1:10] #Kivesszük a kötvényt

r <- mean(r[-11]) #kivesszük a kötvény hozamát

opt\_portfolio<-portfolio\_fv(data,r)

#Optimális portfólió súlyai

w\_opt<-round(opt\_portfolio$weights\*100,2);w\_opt

#Optimális portfólió hozama

r\_opt<-opt\_portfolio$return;r\_opt

#Optimális portfólió kockázata

risk\_opt<-sqrt(opt\_portfolio$weights%\*%cov(data)%\*%opt\_portfolio$weights);risk\_opt

points(risk\_opt,r\_opt,col="red",pch=16)

text(risk\_opt,r\_opt,"opt",c(1,0))

## Hatékony portfóliók görbéje

#Azaz az adott kockázati szinten a legnagyobb hozamú portfólió!

eff\_port\_fv <- function(data)

{

n <- ncol(data)

ri <- colMeans(data)

p<- 3\*n

r <- seq(min(ri), max(ri), length=p)

weights <- rep(0, n)

weights[which.min(ri)] <- 1

for (i in 2:(p-1)) {

newWeights <- portfolio\_fv(data, r[i])$weights

weights <- rbind(weights, newWeights)} #soronként

newWeights <- rep(0, n)

newWeights[which.max(ri)]<- 1

weights <- rbind(weights, newWeights)

weights <- round(weights, 4)

colnames(weights) = colnames(data)

rownames(weights) = 1:p

list(weights = weights, r = r)

}

eff\_port<-eff\_port\_fv(data)

#A különböző hozamokhoz tartozó optimális portfóliók súlyai

weights <- eff\_port$weights;weights

apply(weights,1,sum) #ellenőrzés, sorösszeg 1

r\_eff<-eff\_port$r;r\_eff

#A különböző hozamokhoz tartozó optimális portfóliók kockázatai

risks <- NULL

for (i in 1:nrow(weights)) {

new\_risk <- sqrt(weights[i, ] %\*% cov(data) %\*% weights[i, ])

risks <- c(risks, new\_risk)

}

risks

#Hatékony portfóliók görbéje

points(risks, r\_eff,type="l")

##PÉLDA### Swiss pension fund assets returns benchmark - Hozamok a Svájci nyugdíjalapoknál

install.packages('fBasics')

library(fBasics)

data <- 100 \* LPP2005REC[, 1:6]; data[1:10,] #100\* az LPP2005REC adatsor 1-6 oszlopai

plot(LPP2005REC[, 1:6])

# PORTFÓLIÓ ELVÁRT HOZAMA (oszopátlagok átlaga)

r <- mean(colMeans(data))

#AZ OPTIMÁLIS PORTFÓLIÓ

portfolio <- portfolio\_fv(data,r);portfolio

#EREDMÉNY

weights <- portfolio$weights\*100;weights #súlyok kiiratása

sum(weights) #súlyok összege 100

c(weightedReturn = round((weights %\*% colMeans(data))[[1]],3), r = round(100 \* r, 3)) #ELLenőrzés

Weights <- weights[weights > 0] #csak a pozitív súlyúak

pie(Weights, labels = names(Weights),main="LPP2005 Portfolio Weights")

r\_opt<-portfolio$return

risk\_opt<-sqrt(portfolio$weights%\*%cov(data)%\*%portfolio$weights)

eff\_port<-eff\_port\_fv(data)

weights <- eff\_port$weights;weights

r<-eff\_port$r;r

risks <- NULL

for (i in 1:nrow(weights)) {

new\_risk <- sqrt(weights[i, ] %\*% cov(data) %\*% weights[i, ])

risks <- c(risks, new\_risk)

}

risks

plot(risks, r,type="l",xlab="Kockázat",ylab="Hozam",main="Hatékony portfóliók görbéje")

points(risk\_opt,r\_opt,col="red",pch=16)

text(0.18,0.05,"(0.24,0.043)")